TP 1

Estimation et compensation du mouvement

Vous travaillerez en partie binômes, en partie tous ensemble.

1 Matériel à disposition

Sur le site :

http://perso.telecom-paristech.fr/~cagnazzo/p180.php

Vous disposez de:

- Plusieurs fichiers vidéo (.yuv et .y) et un fichier contenant des vecteurs de mouvement (.mat)
- Un programme de visualisation (seqview.exe)
- Plusieurs fonctions Matlab :
  - Fonctions de lecture/écriture des images d'une vidéo (readFrame.m, appendFrame.m)
  - Fonction d'affichage des images et des vecteurs de mouvement (showFrame.m, displayMVF.m)
  - Fonction de calcule du PSNR
  - Fonction de compensation du mouvement (mc.m)
  - Squelette de la fonction de block matching (bm.m)

2 Familiarisation avec les outils

Utilisez le matériel disponible pour :

1) Lire les vidéos en utilisant seqview. La suite du TP se fait **en Matlab.**
2) **Utiliser la commande *help* pour connaître le mode d'emploi des fonctions que vous serez amenés à utiliser (p.e., tapez *help readFrame*)**
3) En Matlab, lire et afficher un' image d'une vidéo en niveaux de gris. Stocker l'image dans la variable *grayFrame.*
4) Lire et afficher un' image d'une vidéo en couleur. Stocker l'image dans la variable *colorFrame.*
5) Avec la commande *whos* comparer les variables *grayFrame* et *colorFrame*.
6) Lire et afficher les vecteurs de mouvement de l'image 10 (relativement à la 9) pour la vidéo **mobile** (cliquer sur le fichier .mat, et utiliser displayMVF). Commenter les caractéristiques de la variable *mvf*
7) En Matlab, effectuer la compensation du mouvement de cette image, et afficher le résultat avec la commande *showframe*
8) Calculez l'image d'erreur (différence entre l'image 10 et la version compensée de l'image 9). Affichez l'image d'erreur avec *showFrame.* Affichez aussi l'échelle des couleurs avec la commande *colorbar.* Commentez sur la position des erreurs les plus grandes.
9) Après avoir compris la fonction *psnr.m*, qui calcule le PSNR entre 2 images, calculer le PSNR entre l'image 10 et l'image 9 compensée, et entre 10 et 9 non compensée. Commenter sur l'efficacité de la compensation.
10) Créer un fichier vidéo formé par les premieres 9 images de *mobile*, l'image 9 compensée en mouvement, et les images de 11 à la fin (utilisez *appendFrame.m*). Visualisez-le avec seqview.

3 Implémentation du block matching

1. Utiliser le squelette bm.m pour créer une fonction qui fait l'estimation du mouvement par block matching. D'abord vous pouvez travailler tous ensemble pour compléter *bm.m*.
2. Appliques *bm.m* sur des couples d'images consécutives des séquences B/N ou couleurs.
3. Utilisez *mc.m* pour calculer l'image compensée et l'erreur de compensation, comme dans 2.6-2.9
4. En suite, les différents binômes peuvent implémenter différents versions du BM, en modifiant :
   - La mesure de similarité (SSD ou SAD, utilisation ou non des composantes de couleur)
   - La stratégie de recherche (Full search, three step, cross search, ...)
   - La régularisation (pénaliser la différence du vecteur courant par rapport à la moyenne, médiane des voisins ; pénaliser le vecteur courant par rapport à la valeur absolue du gradient, …)

4 Evaluation et comparaison des techniques

1. Chaque binôme évalue la complexité des techniques implémentées en calculant le temps d'exécution de sa version de la fonction *bm.m*.
2. On évalue aussi la régularité du champ comme somme de la valeur absolue du gradient du champ (voir la fonction gradient de Matlab).
3. Comparez les fonctions d'EM à l'aide du script compare.m
4. Commentez les résultats.

5 Estimation du mouvement dense

Modifiez la fonction *bm.m* pour faire en sorte de produire un vecteur différent par pixel. Vous pouvez d'abord réduire la taille des blocs. Quel est l'effet sur la régularité ? En suite il faudra utiliser des blocs avec superposition.

Conclusion

À la fin du TP vous aurez plusieurs version de *bm.m* (que vous appellerez avec des noms differents), dont vous connaitrez les *trade-off* en termes de PSNR, densité, régularité et temps de calcule.